07 – reinforcement learning

RL is mapping of X to Y in order to maximise Z. What are X, Y & Z? Actions, situations, reward signalWhat does RL explicitly consider? The whole problem of a goal-directed agent interacting with an unknown environment ON07 – REINFORCEMENT LEARNING SPECIFICALLY FOR YOUFOR ONLY$13. 90/PAGEOrder NowWhat is a less formal way of describing RL? Learning through trial & errorIn RL, what are the two types of feedback the environment provides the agent? State & state evaluation (reward)In RL, what signal does the agent send to the environment? ActionIn RL what does the reward signal communicate? What is required, not how to do itIn RL what does the reward signal tell the agent about the action's correctness? Does not explicitly indicate whether action was correct or incorrectWhat are the 3 main differences in supervised learning vs reinforced learning? 1.) The system learns from examples by a knowledgeable external factor 2.) Environment explicitly indicates what the agent's action should have been 3.) Instructive feedback independent of outputWhat are the 2 main situations where a RL system of learning is appropriate? 1.) When it is impossible to get sufficient examples of desired behaviour 2.) Learning from experience becomes more appropriate when it becomes difficult for the examples of desired behaviour correct and representative of all situations the agent is likely to experienceWhat are the 3 main components of an RL algorithm? 1.) Reward function 2.) Value function 3.) PolicyWhat is the " reward function"? Function that defines a goal by specifying a number for each state-action combinationWhat is the " value function"? Function that specifies total reward expected when starting from a given state with a given behaviourWhat is the " policy"? Mapping from perceived state to actionWhat is " discounting"? Reward is discounted over longer runs according to some

discount rate with range 0 - 1What is the discounted return R at time t (discount function)? (PHOTO)R▼t = r▼t+1 + xr▼t+2 + xr▼t+3... = Σx^nr▼t+nWhat is the action value estimation function (for small k values where maintaining a prior reward list is feasible)? (PHOTO)Q▼t(a) = (r▼1 + r▼2... + r▼k) / kWhat is the action value estimation function (for large k values where maintaining a prior reward list is impossible)? (PHOTO)Q▼k+1 = Q▼k + 1/(k+1) x (r▼k+1 - Q▼k)

or replace 1/(k+1) with a constant for dynamic tasksDescribe a greedy reward policyHighest action value used to select output for given situationName 2 other reward policies besides " greedy" ε-greedy, annealedWhat two things does an environment with the Markov Property allow us to predict? Its one-step dynamics enable next-state predictions & expected next rewardCompare policy for a Markov state vs. policy as a function of complete historiesThey are the sameExplain partially-observable states (the perceptual aliasing problem)If an entity's inputs convey partial information about the environment, there may be situations which appear identical to the agent but require different optimal actionsSARSA PSEUDOCODE PHOTOSARSA PSEUDOCODE PHOTOExplain " on-policy" An on-policy algorithm evaluates the policy actually usedExplain " off-policy" An off-policy algorithm approximates optimal action-value function independently of the policy being followedONE-STEP Q-LEARNING FORMULA PHOTOONE-STEP Q-LEARNING FORMULA PHOTOWhat are the 6 main steps in Q-Learning Pseudocode (PHOTO)1.) Initialise action value function for states s and actions a Q(s, a) 2.) For each episode... Initialise s 3.) For each step of trial... Choose a from s via policy derived from Q 4.) Take action a, observe r, s' 5.) Update Q(s, a), 6.) s < -s' END foreach END foreach UNTIL terminalWhy is

creating a table value for every state-action mapping not feasible in some problems? In complex problems, the data memory required or time to visit all combinations would be too largeHow do you generalise in complex problems? Approximate for states not experienced, through supervised learningWhat are two ways you can approximate functions during generalisation of complex problems? Gradient descent, MLPsBriefly describe how you would use an MLP to approximate functions in complex problem generalisationUse on MLP per action, use the state as the input, MLP returns Q(s, a) per stepGive two applications of RL in game-playingBackgammon, draughtsGive two applications of RL in engineeringRobotics, lift-allocationGive two applications on RL in softwareAdaptive games, browser agents